REMARKS/ARGUMENTS

Claims 1-3, 5-8, 10-13 and 15 are pending in the present application. Reconsideration of the claims is respectfully requested.

I. 35 U.S.C. § 102, Anticipation

The Examiner has rejected claims 1, 3, and 6-7 under 35 U.S.C. § 102 as being anticipated by U.S. Patent Application Publication 2002/0095553 published by *Mendelson et al.,* (hereinafter *Mendelson*). This rejection is respectfully traversed.

Applicants' claim 1 describes a first cache for storing a first plurality of instructions and a second cache for storing a second plurality of instructions. Each instruction of the first plurality has an associated counter. When a first instruction of the first plurality is accessed, its first associated counter is incremented.

When the first associated counter reaches a threshold, the first instruction is copied into the second cache. The first instruction is copied when the first associated counter reaches a threshold. As is clear from the claim language, the copying is not discretionary. The copying of the first instruction is required when the first associated counter reaches the threshold.

The claim also clearly defines when that required copying must occur. The first instruction is copied when the first associated counter reaches the threshold. According to claim 1, the copying occurs at the time the counter merely reaches the threshold. Thus, copying occurs prior to the counter exceeding the threshold.

Mendelson teaches a filter trace cache (FTC) and a main trace cache (MTC). Newly received traces are placed in the FTC. At a later time, a trace may be, but is not required to be, evicted from the FTC. At that time, the evicted trace will either be discarded or moved to the MTC. The decision as to whether to discard a trace or move it to the MTC will depend on that trace's access counter value. If the value is below a threshold, the trace is discarded. If the value is equal to or above a threshold, the trace is moved to the MTC.

Traces can stay in the FTC indefinitely regardless of their counter value. Traces stay as long as new traces do not evict them. See page 3, paragraph 0033. Thus, a trace stays in the FTC until a new trace is received that must replace the existing trace. If, for example, a trace is in the FTC and no new traces are received, that existing trace will stay in the FTC.

A trace will stay in the FTC regardless of its counter value. Thus, the trace in the example above where no new traces are received will stay in the FTC while its counter value continues to increase well beyond the threshold value. A trace is only evicted from the FTC when a new trace needs to be stored in

the existing trace's location. Only when a new trace is received that needs to replace an existing trace is a determination made as to whether to discard the existing trace or to move the existing trace to the MTC.

Traces are not moved out of the FTC because their values reached a threshold. Traces can stay in the FTC even after their counter values reached a threshold. Traces are moved out only because a new trace will be stored in the existing trace's location in the cache.

A decision as to whether to discard the existing trace or move the existing trace requires evaluating the counter. This decision is only made once an existing trace is evicted. This decision is not made prior to the trace being evicted. Prior to the trace being evicted, the trace remains in the FTC regardless of the counter value. The counter can continue to increase indefinitely, well beyond the threshold.

No action is taken to move a trace out of the FTC based on the trace's counter value. A trace is only moved out of the FTC when that trace is evicted by a new trace. The counter value plays no part in the decision as to whether or not a trace will be evicted.

Applicants' claim 1 describes "when the first associated counter reaches a threshold, the first instruction of the first plurality is copied into the second cache". Thus, according to Applicants' claim, the first instruction is copied when the counter reaches a threshold. No other action is involved. When the counter reaches the threshold, the first instruction is copied. Unlike *Mendelson* where traces can stay in the FTC after their counter value reaches a threshold, according to Applicants' claims, the first instruction cannot stay in the first cache after its counter reaches a threshold. The first instruction must be copied once its counter reaches the threshold.

Mendelson does not teach a trace being copied when its value reaches a threshold. In *Mendelson*, a trace can remain in the FTC with its value well above the threshold. Therefore, *Mendelson* does not teach when the first associated counter reaches a threshold, the first instruction of the first plurality is copied into the second cache. Therefore, *Mendelson* does not anticipate Applicants' claim 1.

Applicants' claim 3 depends from claim 1. Because *Mendelson* does not anticipate claim 1, *Mendelson* does not anticipate claim 3.

Applicants' claim 6 describes if the first associated counter exceeds the threshold, moving the first instruction from the first cache to a second cache. Similarly to claim 1, moving the first instruction is not discretionary. The first instruction is moved if the first associated counter exceeds the threshold.

As discussed above, *Mendelson* teaches a cache, the FTC, in which a trace can remain indefinitely while its value far exceeds the threshold. Traces are not moved out of the FTC because their values exceeded a threshold. Traces are moved out because a new trace will be stored in the existing trace's location in the cache. The trace's counter value is used to determine whether that trace, which is about to be overwritten, will be overwritten (and essentially discarded) or whether that trace will be

moved to the MTC. The trace is not moved because its counter value exceeded the threshold. The trace is either discarded or moved because another trace is about to overwrite the existing trace.

According to Applicants' claims, the first instruction is moved if the counter exceeds the threshold. This is mandatory according to Applicants' claims.

Mendelson does not teach a trace being copied if its value exceeds a threshold. In *Mendelson*, a trace can remain in the FTC even after its value far exceeds the threshold. Therefore, *Mendelson* does not teach if the first associated counter exceeds the threshold, moving the first instruction from the first cache to a second cache. Therefore, *Mendelson* does not anticipate Applicants' claim 6.

Applicants' claim 7 depends on claim 6. Because *Mendelson* does not anticipate claim 6, *Mendelson* does not anticipate claim 7.

II. 35 U.S.C. § 103, Obviousness

The Examiner has rejected claims 2 and 8 under 35 U.S.C. § 103 as being unpatentable over *Mendelson* in view of U.S. Patent Application Publication 2001/0001873 published by *Wickeraad et al.*, (hereinafter *Wickeraad*). This rejection is respectfully traversed.

Claims 2 and 8 describe similar features. These claims describe wherein each instruction of the second plurality has an associated counter, and wherein when an instruction of the second plurality is accessed, all other counters of the second plurality are decremented.

The Examiner relies on *Wickeraad* to cure the deficiencies of *Mendelson*. *Wickeraad*, however, does not cure the deficiencies of *Mendelson*. The combination does not render these claims unpatentable because the combination does not describe, teach, or suggest the combination of wherein when the first associated counter reaches a threshold, the first instruction of the first plurality is copied into the second cache, wherein each instruction of the second plurality has an associated counter, and wherein when an instruction of the second plurality is accessed, all other counters of the second plurality are decremented.

The Examiner has rejected claims 5 and 10 under 35 U.S.C. § 103 as being unpatentable over *Mendelson* in view of Norman P. Jouppi, "Improving Direct-Mapped Cache Performance by the Addition of a Small Fully-Associative Cache and Prefetch Buffers" (hereinafter *Jouppi*). This rejection is respectfully traversed.

Claims 5 and 10 describe similar features. These claims describe the first cache being an instruction cache and the second cache being fully associative and following a least recently used policy.

The Examiner relies on *Jouppi* to cure the deficiencies of *Mendelson*. *Jouppi*, however, does not cure the deficiencies of *Mendelson*. The combination does not render these claims unpatentable because the combination does not describe, teach, or suggest the combination of wherein when the first associated

counter reaches a threshold, the first instruction of the first plurality is copied into the second cache and the first cache being an instruction cache and the second cache being fully associative and following a least recently used policy.

The Examiner has rejected claims 11-12 under 35 U.S.C. § 103 as being unpatentable over *Mendelson* in view of Andrew S. Tannenbaum, "Structured Computer Organization, 2nd Edition" (hereinafter *Tannenbaum*). This rejection is respectfully traversed.

Claim 11 describes fourth instructions for, if the first associated counter exceeds the threshold, moving the first line of data from the first cache to a second cache.

The Examiner relies on *Tannenbaum* to cure the deficiencies of *Mendelson*. *Tannenbaum*, however, does not cure the deficiencies of *Mendelson*. The combination does not render these claims unpatentable because the combination does not describe, teach, or suggest fourth instructions for, if the first associated counter exceeds the threshold, moving the first line of data from the first cache to a second cache.

Claim 12 depends from claim 11 and is believed patentable for these reasons given above.

The Examiner has rejected claim 13 under 35 U.S.C. § 103 as being unpatentable over *Mendelson* in view of *Tannenbaum* and further in view of *Wickeraad*. This rejection is respectfully traversed.

Claim 13 describes wherein each line of data of the second cache has an associated counter, and wherein when a line of data of the second cache is accessed, all other counters of the second cache are decremented.

The Examiner relies on the combination of *Tannenbaum* and *Wickeraad* to cure the deficiencies of *Mendelson*. The combination of *Tannenbaum* and *Wickeraad*, however, does not cure the deficiencies of *Mendelson*. The combination does not render these claims unpatentable because the combination does not describe, teach, or suggest the combination of fourth instructions for, if the first associated counter exceeds the threshold, moving the first line of data from the first cache to a second cache and wherein each line of data of the second cache has an associated counter, and wherein when a line of data of the second cache is accessed, all other counters of the second cache are decremented.

The Examiner has rejected claim 15 under 35 U.S.C. § 103 as being unpatentable over *Mendelson* in view of *Tannenbaum* and further in view of *Jouppi*. This rejection is respectfully traversed.

Claim 15 describes wherein the first cache is an instruction cache and the second cache is fully associative and follows a least recently used policy.

The Examiner relies on the combination of *Tannenbaum* and *Jouppi* to cure the deficiencies of *Mendelson*. The combination of *Tannenbaum* and *Jouppi*, however, does not cure the deficiencies of *Mendelson*. The combination does not render these claims unpatentable because the combination does not describe, teach, or suggest the combination of fourth instructions for, if the first associated counter

exceeds the threshold, moving the first line of data from the first cache to a second cache and wherein the first cache is an instruction cache and the second cache is fully associative and follows a least recently used policy.

III. Examiner's Response to Applicants' Arguments

The Examiner responds, in paragraph 26 of the Final Office Action mailed June 26, 2006, to Applicants' argument that *Mendelson* does not teach "when the first associated counter reaches a threshold, the first instruction of the first plurality is copied into the second cache". The Examiner states that Applicants are relying on features that are not recited in claim 1. Specifically, the Examiner states that it is inherent in the claim that a comparison is made between the first associated counter and a threshold, but the claim does not recite at which point in time this comparison is made.

The Examiner goes on to state that *Mendelson* teaches a comparison that is made when a trace needs to be evicted. At the time a trace needs to be evicted, a comparison is made between a threshold and a number of accesses of that trace. If the number of accesses is equal to or higher than the threshold, the trace is evicted. The Examiner appears to agree that *Mendelson* teaches a situation where a counter of the number of accesses could exceed a threshold.

Mendelson teaches that traces are stored in a filter trace cache (FTC). An existing trace has an access counter that indicates the number of times that trace has been accessed. This counter is permitted to increase beyond a threshold with no action being taken. If there is no new trace that needs to be stored in the FTC, this counter increases indefinitely, possibly far beyond the threshold.

Mendelson then teaches a two-step process. An event must first occur. The occurrence of this event sets a time at which a comparison is made. Then, upon the occurrence of this event, a comparison is made between the counter value and a threshold. The event is the need to store a new trace in the FTC. If a new trace needs to be stored, the counter value is compared to the threshold. If the counter value is equal to or exceeds the threshold, the trace will be moved to a main trace cache (MTC). Therefore, the existing trace is not moved when its counter exceeds the threshold. The existing trace is only moved if a new trace needs to be stored and the counter value is equal to or exceeds the threshold.

When the Examiner states that Applicants' claims do not specify a particular time when a comparison is made, the Examiner is attempting to read a limitation in Applicants' claim language that does not, and need not, exist. The Examiner is attempting to read an event that must occur that then sets a time at which a comparison is made.

Applicants do not claim, either explicitly or implicitly, a two-step process. Applicants do not claim a particular event that must occur before a comparison is made. According to Applicants' claims, only one event must occur. That event is the occurrence of the first counter reaching a threshold. When

the first associated counter reaches a threshold, the first instruction is copied. The first associated counter could reach the threshold at any time. At whatever time the first associated counter reaches a threshold, the first instruction is copied. There is no event that must occur before a comparison is made. According to Applicants' claims, the first instruction is copied upon the occurrence of the first counter reaching a threshold.

Mendelson does not teach an instruction being copied when its counter reaches a threshold. In fact, *Mendelson* teaches away from Applicants' claims because *Mendelson* quite clearly describes a counter exceeding a threshold before it is copied. In *Mendelson*, a counter can increase indefinitely with its associated trace never being copied. Therefore, *Mendelson* does not anticipate Applicants' claims.

IV. Conclusion

It is respectfully urged that the subject application is patentable over the cited prior art and is now in condition for allowance.

The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: August 11, 2006

Respectfully submitted,

/Lisa L.B. Yociss/

Lisa L.B. Yociss
Reg. No. 36,975
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Attorney for Applicants